

A Nonlinear Digital Filter for Industrial Measurements

Abstract: A simple algorithm is presented for nonlinear filtering of a time series composed of a gaussian component, pulses and steps. The method used is a combination of simple statistical techniques. The main advantage is claimed to be a scheme for adaptation of the filter parameters.

Introduction

Time series arising from observations made on industrial processes sometimes contain components that make it necessary to use a nonlinear filter to extract the signal.

Such problems appeared during development work on a process control project jointly carried out by the Billerud Company and by IBM.

The nonlinearities that will be dealt with are large changes in the signal level during a time interval that is short compared to the time constant of the process. These changes may be a step, which is a lengthy change, and a pulse, which is a short irregularity.

Such components might be the effect of set-point changes or spike noise that is caused by instrumental errors or generated by the physical process. In this case we need a filter that gives us information about the step change as soon as possible but cuts the pulses so that no action is taken unnecessarily.

The a priori knowledge about a set-point change has not been considered here since it would have unnecessarily complicated the structure of the algorithm.

It is possible, of course, to take advantage of such knowledge for example by decreasing the decision lag, taking the time constant for each filtered variable into account. This paper presents such a filter algorithm that is designed heuristically and uses a combination of simple statistical techniques.

The process

The industrial process for which the filter is designed is best described in mathematical terms as the sum of a gaussian time series, steps and spike noise.

The gaussian component $\epsilon(t)$ could be described as the combination of a random walk and white noise, and is defined by the model

$$\epsilon(t) - \epsilon(t-1) = e(t) - ce(t-1), \quad (1)$$

where $e(t)$ is a sequence of normally distributed uncorrelated random variables with zero mean and variance λ^2 .

If the parameter $c = 1$, the model (1) reduces to $\epsilon(t) = e(t)$ while if $c = 0$, (1) describes a pure random walk $\epsilon(t) = \epsilon(t-1) + e(t)$. If we define the step variable $\xi(t)$ and the pulse variable $\zeta(t)$ which equal zero when no steps or pulses are present, we can write a model for the entire process as follows:

$$y(t) = \epsilon(t) + \xi(t) + \zeta(t). \quad (2)$$

Problem formulation

The task of the filter is to detect the steps as soon as possible, to sort out the spike noise, and to smooth the gaussian component.

Our concern about the gaussian component, however, will be to predict it rather than smooth it. There are two reasons for this:

To state a smoothing problem requires assumptions about the signal behind the raw data, about which we may know very little.

We can often state a prediction problem for which the result will be the same as for the smoothing problem.

Thus there will be a filtering problem—to follow the steps and cut the spikes—and a prediction problem.

Because of the nature of the model it is natural to divide the filtering problem into the following two parts:

The author is at the University of Uppsala in the School of Engineering, Department of Physics, Uppsala, Sweden.

- 1) Find a test for the hypothesis that there is no pulse or step present at time t under the assumption that there has been none earlier.
- 2) Find a method to distinguish between a step and a pulse when the hypothesis is rejected.

The first part assumes a gaussian process, and some other technique must be used for another type of stochastic process. The prediction problem concerns the time series that is left when steps and pulses are sorted out. This time series will not be normal since the probabilities of making a wrong decision in the first problem (α - and β -errors²) are nonzero. Thus a linear prediction will not be a conclusion but it will be postulated in order to get a simple algorithm.

- 3) Find a prediction $\hat{\epsilon}(t | t - r)$ of $\epsilon(t)$ which is a linear combination of earlier observations $\epsilon(t - r)$, $\epsilon(t - r - 1) \dots$, and which minimizes the mean square prediction error.

The prediction problem, as stated here, has been thoroughly investigated^{1,4,5} and only a summary of prediction one step ahead will be given since, for our model, the result will be the same for several steps ahead.

Prediction one step ahead

Introducing the translation operator z so that $ze(t) = \epsilon(t - 1)$, the model (1) can be written as

$$\begin{aligned}\epsilon(t) &= \frac{1 - cz}{1 - z} e(t) = e(t) + \frac{(1 - c)z}{1 - z} e(t) \\ &= e(t) + \frac{(1 - c)z}{1 - cz} \epsilon(t)\end{aligned}\quad (3)$$

Since $e(t)$ are uncorrelated, the optimal linear predictor becomes¹

$$\hat{\epsilon}(t | t - 1) = \frac{(1 - c)z}{1 - cz} \epsilon(t) \quad (4)$$

from which expression we get the algorithm:

$$\hat{\epsilon}(t + 1 | t) = c\hat{\epsilon}(t | t - 1) + (1 - c) \epsilon(t).$$

The prediction error can according to (3) and (4) be written:

$$\epsilon(t) - \hat{\epsilon}(t | t - 1) = e(t). \quad (5)$$

The filtering problem

The first part of the filtering problem will be treated by hypothesis testing, as is indicated already in the formulation of the problem. As a test variable the gaussian function $e(t)$ will be chosen. Equation (5) indicates that $e(t)$ equals the prediction error in the optimal linear predictor. Since it is approximately normally distributed with zero mean and variance λ^2 , the critical region will be chosen $|e(t)| > 3\lambda$ to give a level of significance about .25 percent.

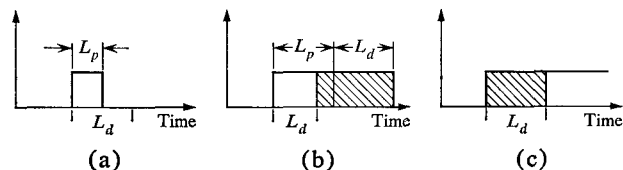


Figure 1 Sampling intervals L_d for detecting pulses and steps. (a) No loss occurs when the pulse length is shorter than the decision lag since the pulse is too short for the filter to react. (b) The pulse length exceeds the decision lag and the loss will now equal the pulse length. (c) When a step occurs, the loss will be equal to the time it takes to decide to follow the step.

If the test is used sequentially, this is correct in the limiting case where the probability of pulse or step tends to zero, since only then the estimated variable $e(t)$ will be normally distributed.

The second problem is a bit more complicated.¹⁻³ Because steps and pulses look very similar except that pulses are of finite duration, the only way of detecting a step is to wait, say L_d sampling intervals, until one can be reasonably sure that $y(t + L_d + 1)$ does not return to the same level as $y(t - 1)$. L_d will be called the *decision lag* and it will be optimized by minimizing the loss suffered from, on the one hand waiting too long to follow a step, and on the other hand following a pulse. Assume an a priori conditional probability P_1 for pulses and P_2 for steps, where $P_1 + P_2 = 1$. Assume further an a priori distribution for pulse length L_p : $P(L_p = k) = p_k$, $k = 1, 2, \dots$ (sample interval as unit). Let L_d be the decision lag. Define a stochastic variable F representing the loss from following a pulse and from not following a step as follows:

$$F(L_d) = \begin{cases} 0 & \text{if } 1 \leq L_p < L_d & \text{(Fig. 1a)} \\ L_p & \text{if } L_d \leq L_p & \text{(Fig. 1b)} \\ L_d & \text{when step} & \text{(Fig. 1c)} \end{cases}$$

If the decision lag is larger than the pulse length (Fig. 1a) there will be no loss since the pulse will be sorted out, but if the pulse length is larger (Fig. 1b), the loss will be equal to the pulse length, and when a step occurs (Fig. 1c), the loss equals the time it takes to make the decision.

The expectation of F becomes:

$$E(F) = P_1 \sum_{L_d}^{\infty} k p_k + P_2 L_d.$$

The problem is now to find a value of L_d that minimizes $E(F)$, which is readily solved, given the a priori distribution.

Example: Assume a geometric distribution of pulse length $p_k = qp^{k-1}$ ($q = 1 - p$). We get

$$\begin{aligned}E(F) &= P_1 \sum_{L_d}^{\infty} k qp^{k-1} + P_2 L_d \\ &= P_1 \frac{p^{L_d}(qL_d + p)}{pq} + P_2 L_d\end{aligned}$$

which expression is a minimum for the smallest value of L_d , giving

$$L_d p^{L_d} - \frac{p}{q} \frac{P_2}{P_1} < 0.$$

The table gives L_d for various values of p and P_1/P_2 :

p	P_1/P_2				
	1	5	20	50	100
.5	1	5	8	9	10
.6	1	4	6	7	8
.7	1	3	5	6	7
.8	1	3	4	5	5
.9	1	2	3	4	4

If the probability ratio between pulses and steps is 50 and the distribution of the pulse length is geometric with 80 percent probability that a pulse is one time unit long, to minimize the loss function five non-normal points should occur in a row before the filter decides that a step has occurred.

The basic algorithm

We now summarize previous discussions and results to get the algorithm.

The first thing we do when reading a new point in the sequence of observations is to determine whether or not a pulse or step is present. To do this, the prediction error is computed by taking the difference between the new point $y(t)$ and the latest estimate $\hat{y}(t | t-1)$. If the magnitude of the result is less than three standard deviations of the normal time series the point is considered as normal. The pulse and step variables $\hat{\xi}(t)$, $\hat{\zeta}(t)$ and the pulse length variable L are then set to zero and the new estimate is given by (4). If the result is larger than three standard deviations, the point is considered as a pulse of the same magnitude as the prediction error and if L_d pulses with the same sign occur in a row it will be considered as a step with the size equal to that of the last pulse. In symbols:

I. Randomization and test of normality

$$\hat{e}(t) = y(t) - \hat{y}(t | t-1)$$

$$a) \hat{e}^2(t) \leq 9\lambda^2$$

$$\hat{\xi}(t) = 0$$

$$\hat{\zeta}(t) = 0$$

$$L = 0$$

$$b) \hat{e}^2(t) > 9\lambda^2$$

$$\hat{s}(t) = \hat{e}(t)$$

$$L = \begin{cases} 1 & \text{if } \hat{\zeta}(t) > 0 \\ -1 & \text{if } \hat{\zeta}(t) < 0 \end{cases} \text{ and } L\hat{\zeta}(t) \leq 0$$

$$L = \begin{cases} L+1 & \text{if } \hat{\zeta}(t) > 0 \\ L-1 & \text{if } \hat{\zeta}(t) < 0 \end{cases} \text{ and } L\hat{\zeta}(t) > 0$$

$$1) |L| < L_d$$

$$\hat{\xi}(t) = 0$$

$$2) |L| \geq L_d$$

$$\hat{\xi}(t) = \hat{\zeta}(t)$$

$$L = 0.$$

II. Filtering and prediction

$$\hat{y}(t+1|t) = c\hat{y}(t | t-1) + (1-c)[y(t) - \hat{\xi}(t)] + \hat{\xi}(t)$$

State variables: $\hat{y}(t+1|t)$, L

Parameters: λ , c , L_d .

On-line adaptation of the parameters

The two parameters c and λ in the basic algorithm both relate to the normal time series and they give a complete description of it. Conversely, the parameters can be estimated using a realization of the time series. If c is known, one way to adapt λ is to compute the mean square prediction error since, according to (5),

$$E[\epsilon(t) - \hat{\epsilon}(t | t-1)]^2 = E\hat{e}^2(t) = \lambda^2.$$

To compute this average we could use the arithmetic mean in recursive form:

$$\hat{\lambda}^2(t) = \left(1 - \frac{1}{t}\right)\hat{\lambda}^2(t-1) + \frac{1}{t}\hat{e}^2(t). \quad (6)$$

This estimate, however, will follow changes in λ very slowly as t becomes large since it weighs all past $\hat{e}^2$ equally. An alternative is to weigh recent values more than the earlier ones. If the terms of a geometric series with unit sum are chosen as the weights the result is:

$$\hat{\lambda}^2(t) = (1-k)\hat{\lambda}^2(t-1) + k\hat{e}^2(t). \quad (7)$$

The two formulas (6) and (7) give an algorithm for estimating λ^2 using (6) when $1 \leq t < 1/k$ and (7) when $t \geq 1/k$.

If we want to adapt both parameters we have to study the statistical properties of the time series more carefully. Considering the model of the gaussian time series (1) we place

$$V(t) = \epsilon(t) - \epsilon(t-1) = e(t) - ce(t-1)$$

and we study the autocorrelation function $R(h)$ of $V(t)$:

$$\begin{aligned} R(0) &= EV^2(t) = \lambda^2(1 + c^2) = 2S \\ R(1) &= E[V(t)V(t+1)] = -c\lambda^2 = -R \\ R(h) &= E[V(t)V(t+h)] = 0 \text{ when } h > 1. \end{aligned} \quad (8)$$

If we estimate the expectations S and R in a similar manner as shown in (7);

$$\begin{aligned} \hat{S}(t) &= (1 - k)\hat{S}(t-1) + \frac{1}{2}k[\epsilon(t) - \epsilon(t-1)]^2 \\ \hat{R}(t) &= (1 - k)\hat{R}(t-1) - k[\epsilon(t) - \epsilon(t-1)] \\ &\quad \times [\epsilon(t-1) - \epsilon(t-2)]. \end{aligned} \quad (9)$$

The equations (8) can be solved for $\hat{c}$ and $\hat{\lambda}^2$

$$\begin{aligned} \hat{c} &= \frac{\hat{S}}{\hat{R}} \left(1 - \sqrt{1 - (\hat{R}/\hat{S})^2} \right) \\ \hat{\lambda}^2 &= \hat{S} (1 + \sqrt{1 - (\hat{R}/\hat{S})^2}). \end{aligned} \quad (10)$$

Since a pulse or a step at time $t-1$ or $t-2$ would ruin the estimates (9) of S and R it is necessary to introduce a new state variable L_a which will tell when $\hat{S}$ and $\hat{R}$ can be updated. L_a will be set equal to three when a pulse occurs and equal to two when a step occurs, since these are the times it takes for the irregularities to come to time $t-3$. Every time a normal point comes in, L_a will be decreased by one.

The introduction of this new state variable is not a serious disadvantage since it does not need much core space in a computer. On a fixed-word-length computer it can be stored in the same word as L and L_d . The equations (10) may sometimes be too complex for computation in an on-line application. Then the following approximation can be made: $\hat{c}$ is replaced by a linear function and $\hat{\lambda}^2$ by its maximum value. Thus we get $\hat{c} = \hat{R}/\hat{S}$ and $\hat{\lambda}^2 = 2\hat{S}$. The effect of these approximations will be a smoother curve as is illustrated by the examples in Figures 2(a) and 3. The full algorithm, including adaptation of both parameters, is shown in Table 1 as a FORTRAN IV subroutine.

Restrictions

A troublesome case in which this algorithm fails is when there is a large sudden change in the variance, larger than the $9\lambda^2$ -limit that is used to detect pulses and steps. This will cause most observations to be sorted out as pulses or to be taken as steps and the estimated variance will recover very slowly.

The relative number of pulses or steps may not be too large. How large it can be depends on how much the parameters c and λ vary with time. In the generated example this number is 4 percent.

Table 1 Full algorithm including adaptation of both parameters.

```

C      ADAPTATION OF BOTH PARAMETERS
C      SUBROUTINE FILIP3 (Y,YBAR,S,R,L,LD,LA)
C      DIMENSION Y(3)
C      CURRENT VALUE OF Y IS IN Y(3)
C
C      RANDOMIZATION AND TEST OF NORMALITY
C      N = 3
C      EBAR = Y(N) - YBAR
C      IF (EBAR*EBAR - 9.*VAR) 200,200,300
C
C      ACCEPT NULLHYPOTHESIS
200  L = 0
C      LA = LA - 1
C      IF (LA-1) 210,212,214
210  R = R - .01*((Y(N)-Y(N-1))*(Y(N-1)-Y(N-2)) + R)
212  S = S + .01*(.5*(Y(N)-Y(N-1))*(Y(N)-Y(N-1)) - S)
214  IF (.95*S-R) 218,220,220
218  R = .95*S
220  CBAR = S/R *(1. - SQRT(1.-R*R/S/S))
C      VAR = S *(1. + SQRT(1.-R*R/S/S))
C      YBAR = CBAR*YBAR + (1.-CBAR)*Y(N)
C      GO TO 400
C
C      REJECT NULLHYPOTHESIS
300  LA = 3
C      IF (FLOAT(L)*EBAR) 310,310,320
310  L = EBAR/ABS(EBAR)
C      GO TO 330
320  L = L + IFIX(EBAR/(ABS(EBAR)))
330  IF (L*L-LD*LD) 400,340,340
340  L = 0
C      LA = 2
C      YBAR = Y(N)
C
C
400  Y(N-2) = Y(N-1)
C      Y(N-1) = Y(N)
C      RETURN
State variables:  Y(t+1|t) = YBAR
                  S(t)    = S
                  R(t)    = R
                  L_a     = LA
                  L        = L
Parameter:       L_d     = LD

```

Conclusions

The performance of the filter algorithm is best illustrated by the samples taken from a paper-making process (Figures 3 and 4). The author's opinion is that the practical difficulties in stating a reasonable criterion for the performance of a filter in the cases illustrated will make any evaluation of such a criterion meaningless. Instead, any filter should be considered "good" if, when applied to data collected from a physical process, it gives acceptable results, i.e., mimics a human being, who has to draw a smooth curve given the sequence of data, one point at a time, and not knowing the future course of the curve. The advantage claimed is thus automatization rather than a particular kind of "optimality."

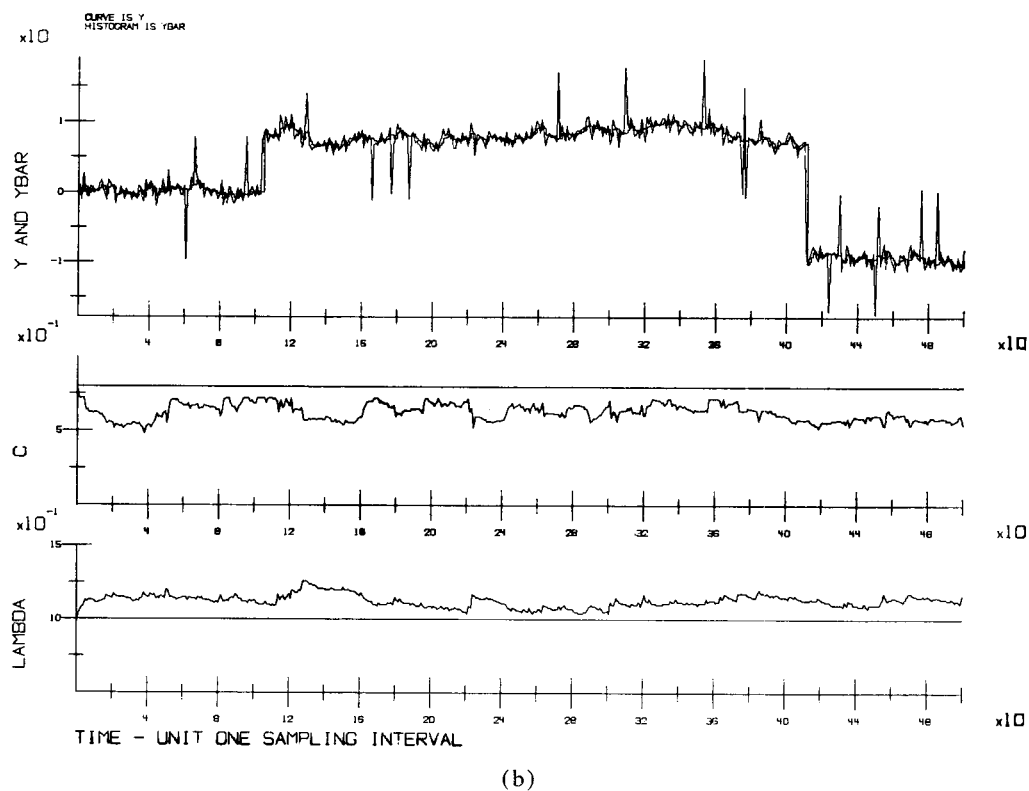
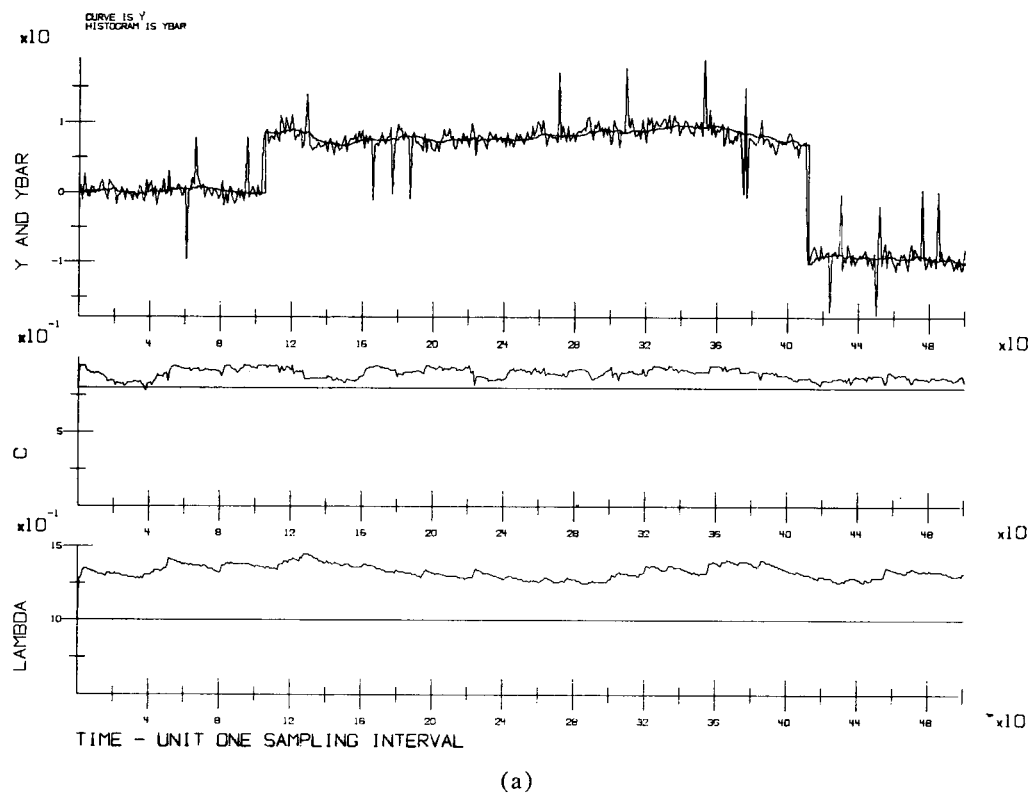
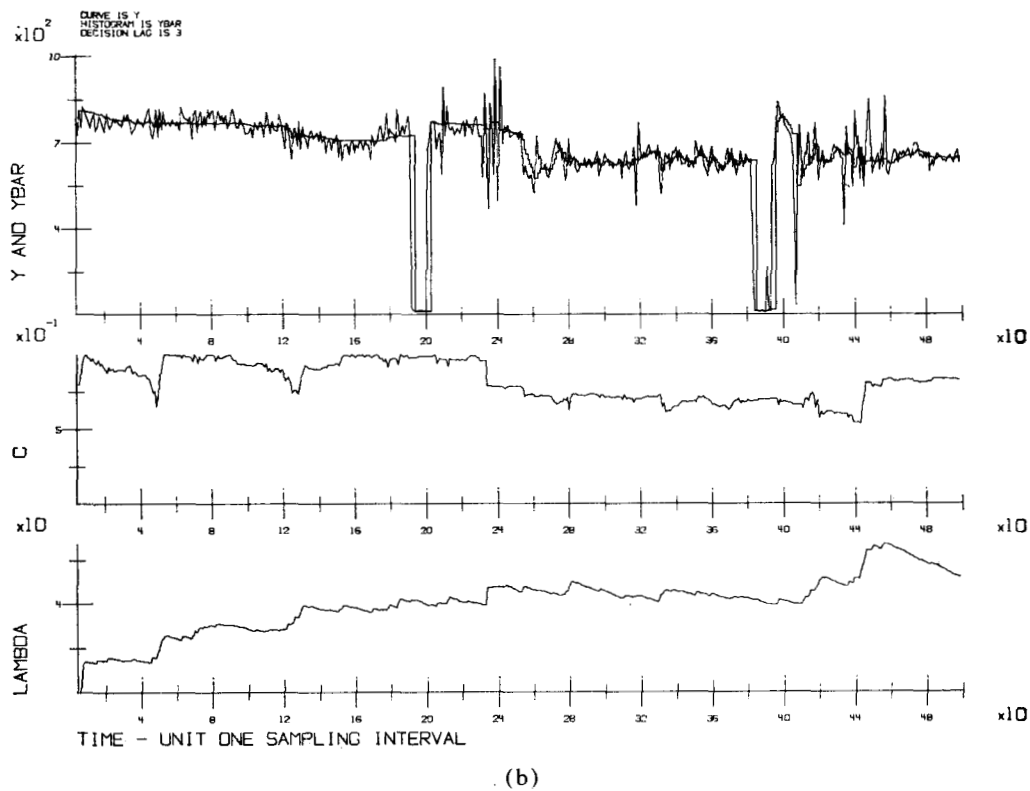
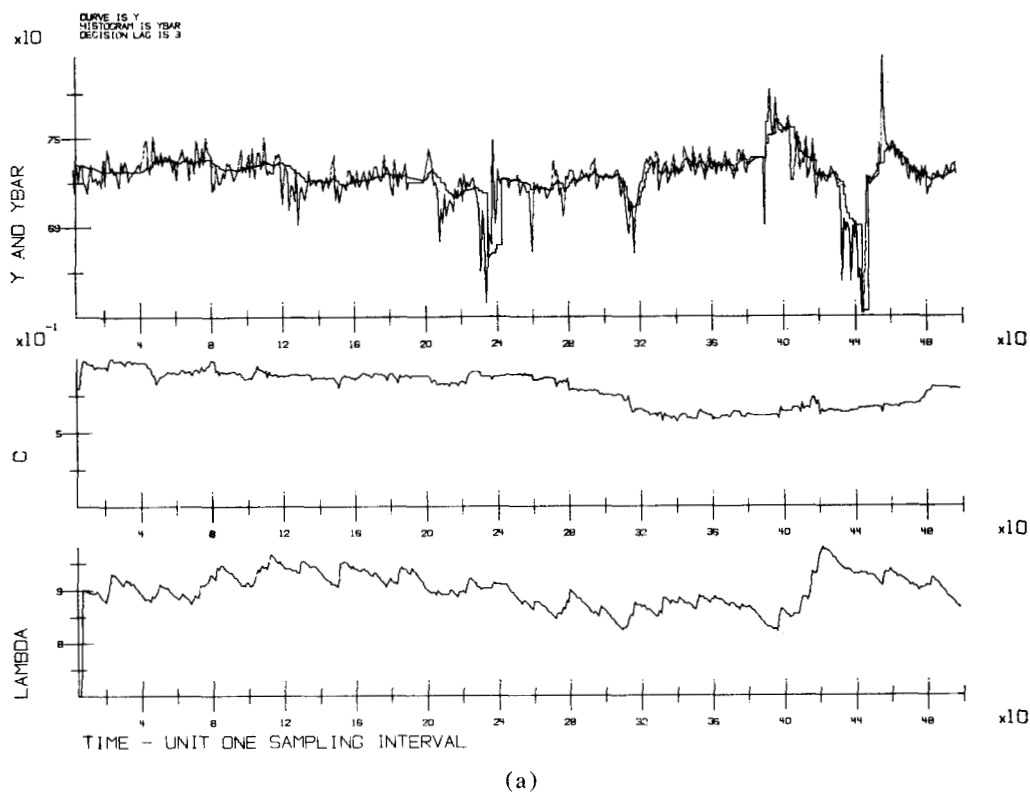


Figure 2(a) Generated process: extra smoothing; (b) adaptation of both parameters.



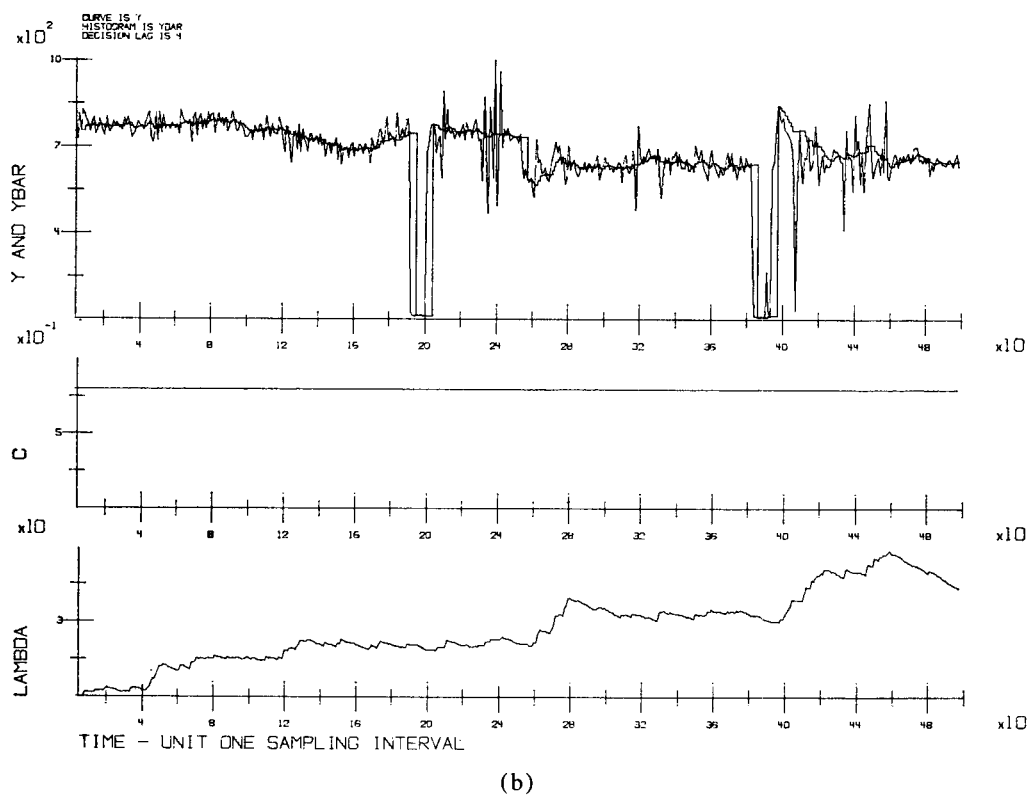
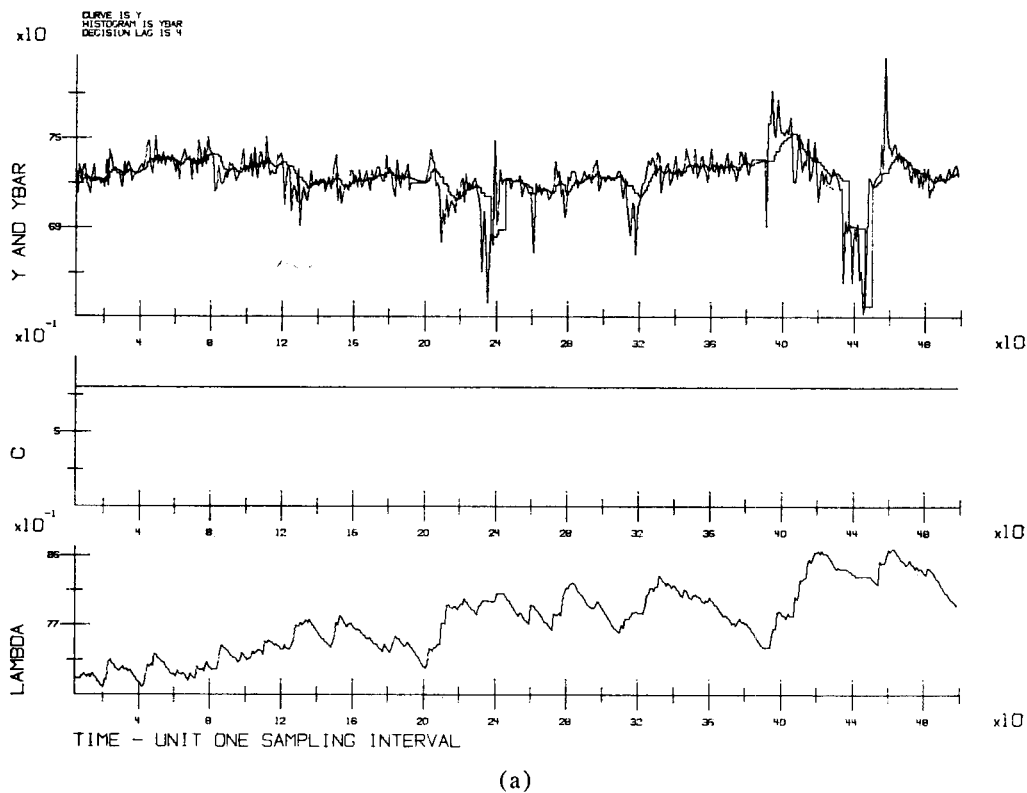


Figure 4(a) Real process, sample 1: adaptation of variance; (b) sample 2: adaptation of variance.

A second, practical, advantage is claimed to be the scheme for adaptation of the filter constants. In effect, the filter algorithm carries out operations similar to the standard delta-check and exponential averaging operations. The latter, however, require constants that are individual for each filtered signal and that the user must assign for each filter input from his process. In practice he may not know the correct values for some or all of (some hundred) inputs, and then his job becomes that much guesswork. If the constants are adapted continuously, the user does not need to specify the constants, and their values remain near optimal as the signal characteristics change.

Acknowledgment

I sincerely want to thank Dr. Torsten Bohlin, IBM Nordic Laboratories, who gave me the problem and who has

assisted me whenever I have had difficulties in my work on it.

References

1. R. Lee, *Optimal Estimation, Identification and Control*, Massachusetts Institute of Technology Press, Cambridge, Mass., 1964.
2. Davenport and Root, *Random Signals and Noise*, McGraw-Hill, Inc., New York, 1958, Ch. 14.
3. R. Deutsch, *Estimation Theory*, Prentice Hall, Englewood Cliffs, N. J., 1965.
4. K. J. Åström, *Notes on the Regulation Problem*, Internal Report CT-209, IBM Nordic Laboratory, Lidingö, Sweden, August 30, 1965, p. 19.
5. K. J. Åström, "Computer Control of a Paper Machine—an Application of Linear Stochastic Control Theory," *IBM J. Res. Develop.* **11**, 389 (1967).

Received January 30, 1968

Revised manuscript received March 12, 1969